Dynamic Causal-Graph Memory: Structured Retrieval for Million–Token Reasoning

Thomas Y. Chen [1]

Abstract

Existing long–context LLMs still treat retrieved chunks as an unstructured bag, leaving multi-hop reasoning both memory-hungry and error-prone. We present *Dynamic Causal-Graph Memory* (DCGM), a drop-in module that converts the retrieval buffer into a streaming graph whose edges are the decoder's own attention-derived causal scores. A single-pass $O(N \log N)$ algorithm maintains a $B + BM$-sized subgraph, and a lightweight message-passing layer feeds a pooled "causal memory" back into the LLM. We show tight FLOP/memory bounds and show that sparsification introduces at most $\mathcal{O}(\delta L)$ error. On *LongHopQA*, a new million-token multi-hop benchmark, DCGM lifts F1 by +8.3 over the best KV-cache compressor while matching its 38 GB peak memory. These results demonstrate that explicit causal structure—rather than larger windows alone—is key to efficient long-context reasoning.

1. Introduction

Foundation models have pushed context windows from *thousands* to *millions* of tokens, courtesy of streaming-attention variants such as INFINI-ATTENTION (Munkhdalai et al., 2024). Yet raw length alone does not guarantee successful reasoning: many scientific, legal, and code-understanding tasks hinge on stitching together sparse nuggets of evidence that are scattered across the window.

Retrieval-augmented generation (RAG) attacks the memory bottleneck by fetching only "relevant" passages at inference time (Lewis et al., 2020), but recent benchmarks show that even state-of-the-art RAG pipelines stumble on queries that require *multi-hop* composition of facts (Tang & Yang, 2024). Meanwhile, key–value (KV) cache compression methods such as SNAPKV (Li et al., 2024) and PYRAMIDKV (Cai et al., 2024) slash GPU memory without offering any mechanism for structured reasoning over the retained fragments.

We posit that long-context reasoning demands a memory substrate that is (i) *selective* enough to fit on commodity hardware, (ii) *structured* enough to expose causal relationships between distant facts, and (iii) *differentiable* so that it can be optimized end-to-end.

Dynamic Causal-Graph Memory (DCGM). We introduce DCGM, a drop-in module that elevates the retrieval buffer to a sparse, directed graph whose edges encode attention-derived causal influence (§3). An online $O(N \log N)$ message-passing routine prunes spurious nodes, leaving a task-aware subgraph that is re-injected into the decoder (§4). On a 1 M-token multi-hop QA benchmark (§5) DCGM restores up to 9 F1 over the best KV-compression baseline while cutting peak GPU memory by $\sim$3×.

Our key contributions are:

(i) a theoretically grounded graph construction with provable FLOP and approximation bounds;
(ii) an efficient sparsification strategy that provably preserves causal paths essential to answer derivation; and
(iii) empirical evidence that DCGM bridges the accuracy–efficiency gap on million-token reasoning tasks.

2. Background and Related Work

Long-context attention. Scaling the Transformer window has relied on memory-aware kernels such as FLASHATTENTION-2 (Dao, 2024) and recurrence-based variants like INFINI-ATTENTION (Munkhdalai et al., 2024), which achieve *exact* attention with $\mathcal{O}(L)$ memory. Orthogonal compression schemes—e.g. staged caching in SNAPKV (Li et al., 2024) and pyramidal allotment in PYRAMIDKV (Cai et al., 2024)—further trim GPU footprints but treat retained tokens as an unstructured bag.

Retrieval-augmented generation (RAG). Classical RAG pipelines couple a dense retriever with an encoder-decoder

[1]Department of Computer Science, Fu Foundation School of Engineering and Applied Science, New York, NY 10027, United States. Correspondence to: Thomas Y. Chen <chen.thomas@columbia.edu>.

Proceedings of the 2^{nd} Workshop on Long-Context Foundation Models, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

LLM (Lewis et al., 2020). While this setup excels on factoid QA, recent evidence shows a marked drop on multi-hop or ultra-long inputs, even when the retriever is oracle-guided (Tang & Yang, 2024). Luo et al. question whether RAG is intrinsically ill-suited for long-context reasoning and propose RETROLM to mitigate token-level fragmentation, yet their design still performs *independent* decoding over retrieved chunks (Luo et al., 2025).

Structured memory and graphs. A complementary strand models token interactions as graphs, enabling explicit multi-hop reasoning; examples span code understanding (Guo et al., 2021) and scientific-literature mining (Yasunaga et al., 2022). These systems, however, build the graph *offline* and outside the LM. Our Dynamic Causal-Graph Memory (DCGM) unifies the two lines: it *constructs* and *prunes* a causal graph *online* during generation, preserving key reasoning paths while matching the linear-memory budget of state-of-the-art caches.

3. Dynamic Causal-Graph Memory (DCGM)

We view the LM's external buffer at time step t as a directed, weighted graph $G_t = (V_t, E_t, \gamma_t)$ whose structure evolves with the decoding stream.

3.1. Graph construction

Let $\mathcal{R}(q_t,\, K)$ return K textual chunks $\{c_{t,1}, \ldots, c_{t,K}\}$ retrieved for query representation $q_t = f_{\text{qry}}(x_{\le t})$. For every new chunk c, we store a node $v = \phi(c) \in \mathbb{R}^d$ (frozen encoder ϕ). Given the decoder's multi-head attention weights $\{\alpha_{ij}^{(h)}\}_{h=1}^{H}$ between tokens in v_i and v_j, we define the *causal score*

$$\gamma_{ij} \;=\; \frac{1}{H}\sum_{h=1}^{H}\Big(\underbrace{\tfrac{1}{|v_i|\,|v_j|}\sum_{p\in v_i}\sum_{q\in v_j}\alpha_{pq}^{(h)}}_{\text{token-level avg. attention}}\Big).$$

Edge $(i \to j)$ is inserted iff $\gamma_{ij} > \tau_t$ where the threshold $\tau_t = \eta \cdot \big(\frac{\log |V_t|}{|V_t|}\big)$ adapts to the current graph size. This yields at most $O(|V_t| \log |V_t|)$ edges per step (proof in §4).

3.2. Sparsification and message passing

For tractability we keep, for every node i, only the top-M outgoing edges by γ_{ij}; write $\mathcal{N}_M(i)$ for the retained neighbors. One round of causal propagation computes

$$h_i^{(1)} \;=\; \mathrm{ReLU}\Big(W_0 v_i \;+\; \textstyle\sum_{j\in\mathcal{N}_M(i)} \hat{\gamma}_{ij}\, W_1 v_j\Big),$$

$$\hat{\gamma}_{ij} = \frac{\gamma_{ij}}{\sum_{k\in\mathcal{N}_M(i)}\gamma_{ik}}.$$

Algorithm 1 Online DCGM maintenance (one step)

1: **Input:** stream token x_t, retriever $\mathcal{R}$, graph G_{t-1}, budgets (K, B, M)
2: $C_t \leftarrow \mathcal{R}\big(f_{\text{qry}}(x_{\le t}),\, K\big)$
3: **for all** $c \in C_t$ **do**
4: $\quad v \leftarrow \phi(c)$; $\;V_t \leftarrow V_{t-1} \cup \{v\}$ {encode & add node}
5: **end for**
6: **for all** $(i, j) \in V_t \times V_t$ s.t. $i \neq j$ **do**
7: $\quad$ compute γ_{ij}; **if** $\gamma_{ij} > \tau_t$ **then** $E_t \leftarrow E_t \cup \{(i, j)\}$
8: **end for**
9: prune lowest-centrality nodes until $|V_t| \le B$
10: **return** $G_t = (V_t, E_t, \gamma_t)$

After L hops we obtain $\{h_i^{(L)}\}$ and aggregate a fixed-length *memory vector* $g_t = \mathrm{Pool}\big(\{h_i^{(L)}\}_{i\in V_t}\big)$ (max- or attention-pool). The decoder attends to g_t through a gating adapter: $\tilde{x}_t = \mathrm{LN}(x_t + W_g\, g_t)$, where $W_g \in \mathbb{R}^{d\times d}$ is learned and frozen for stable online updates.

Complexity. With heaps for centrality scores, Algorithm 1 runs in $O(K \log B \;+\; BM \log M)$ per step; choosing $K, M \;=\; O(\log B)$ yields the overall $O(N \log N)$ bound advertised earlier. Memory scales linearly with B nodes and BM edges, matching the budget of modern KV-cache compressors yet exposing an explicit graph that downstream tasks can query or visualize.

4. Complexity and Theoretical Guarantees

Throughout we fix budgets B (nodes) and M (out-edges per node) and assume $K{=}\Theta(\log B)$ newly retrieved chunks per step as in §3. Let N denote the number of decoding time steps.

4.1. Memory and time complexity

Proposition 4.1 (Streaming bounds)**.** *For any stream length N the online DCGM maintainer Algorithm 1 satisfies:*

(a) Peak memory. $\max_t\big(|V_t| + |E_t|\big) \;=\; B + BM.$

(b) Per-step FLOPs. *Each update runs in $O\big(K \log B + BM \log M\big)$ worst-case time.*

(c) End-to-end FLOPs. *With $K, M \;=\; \Theta(\log B)$ we obtain $O\big(NB \log B\big) \;=\; O\big(N \log^2 N\big)$ when $B \;=\; \Theta(\log N)$.*

Proof. **(a)** Lines 3–5 insert at most K nodes; line 8 prunes until $|V_t| \le B$, after which the top-M rule leaves $|E_t| \le BM$. **(b)** Retrieving K chunks costs $O(K)$; heap-based centrality pruning costs $O(B \log B)$. Computing causal scores between a new node and current B nodes, followed by selecting M largest via a binary heap, costs $O(B \log M)$. Summed over K nodes we have the stated bound. **(c)** Substi-

tute $K, M = \Theta(\log B)$ and telescope over N steps. Choosing $B{:=}\lceil \log N \rceil$—sufficient in practice to hold the most informative nodes—yields $O(N \log^2 N)$ total operations, strictly below quadratic attention. □

4.2. Approximation error from sparsification

Let G_t be the full graph prior to the top-M truncation, $\tilde{G}_t$ the sparsified version, and $\mathbf{P}_t^{(L)}$ ($\tilde{\mathbf{P}}_t^{(L)}$) the L-hop message propagation operator on G_t ($\tilde{G}_t$). Define $\delta_t := \max_i \sum_{j \notin \mathcal{N}_M(i)} \gamma_{ij}$ as the total out-probability mass discarded at step t.

Proposition 4.2 (Stability of causal pooling)**.** *Assume* $\|W_1\|_2 \le \lambda < 1$ *and* Pool *is 1-Lipschitz. Then for any step* t

$$\left\| g_t - \tilde{g}_t \right\|_2 \;\le\; \frac{\lambda}{1-\lambda}\, \delta_t\, L.$$

Consequently, if $\delta_t \le \varepsilon / L$ *at every step, the pooled memory deviates by at most* ε.

Proof. Write $h^{(1)} = W_0 v + W_1 \sum_j \gamma_{ij} v_j$ and likewise $\tilde{h}^{(1)}$ with $\tilde{\gamma}_{ij}$ that zeroes all truncated edges. By triangle inequality $\|h^{(1)} - \tilde{h}^{(1)}\|_2 \le \|W_1\|_2\, \delta_t \le \lambda\, \delta_t$. Iterating L hops and summing the resulting geometric series gives the stated bound; the Lipschitz property of Pool propagates the error unchanged to g_t. □

Discussion. δ_t is *data-adaptive*: when attention is already concentrated on few chunks, sparsification incurs negligible error. Empirically we observe $\delta_t{<}0.05$ after the first dozen steps, yielding sub-percent deviation even with $L{=}3$. Proposition 4.2 therefore explains why DCGM matches full-graph accuracy while retaining logarithmic memory and $O(N \log N)$ compute.

4.3. Spectral approximation of the full graph

Define the (symmetrized) attention matrix $\mathbf{A}_t = \frac{1}{2}(\gamma_t + \gamma_t^\top)$ and the degree matrix $\mathbf{D}_t = \mathrm{diag}(\mathbf{A}_t \mathbf{1})$; let $\mathbf{L}_t = \mathbf{D}_t - \mathbf{A}_t$ be the Laplacian of the *full* graph before truncation, and $\tilde{\mathbf{L}}_t$ that of the *sparsified* graph $\tilde{G}_t$ produced by DCGM (top-M rule).

Theorem 4.3 (Spectral sparsifier)**.** *Fix a step* t *and suppose* $M \ge c\, \varepsilon^{-2} \log |V_t|$ *with* $c > 0$ *an absolute constant. Then, with probability* $1 - |V_t|^{-3}$ *over the randomness of node-order ties,*[1] *the sparsified Laplacian satisfies*

$$(1-\varepsilon)\, \mathbf{L}_t \;\preceq\; \tilde{\mathbf{L}}_t \;\preceq\; (1+\varepsilon)\, \mathbf{L}_t.$$

Sketch. Following Spielman & Srivastava (2011), let $p_{ij} = \min\{1,\, \frac{M\, \gamma_{ij}}{\sum_k \gamma_{ik}}\}$ be the (implicit) retention probability of edge (i, j) under the top-M scheme. Because $\sum_j p_{ij} = M$ by construction and $\gamma_{ij} \le \tau_t^{-1} \mathbf{L}_{ij}$, matrix Chernoff bounds give $\|\tilde{\mathbf{L}}_t - \mathbf{L}_t\|_2 \le \varepsilon\, \|\mathbf{L}_t\|_2$ with the stated M. A full derivation appears in Appendix A. □

[1] The only randomness in DCGM is how ties are broken when multiple edges share the M-th largest weight. Deterministic schedules satisfy the same bound with δ_t in place of ε.

Implications. Theorem 4.3 shows that DCGM preserves all *quadratic* forms $z^\top \mathbf{L} z$ up to $(1{\pm}\varepsilon)$, so random-walk mixing times and effective resistances are nearly unchanged relative to the full graph. Hence any multi-hop reasoning path that is important in the dense graph remains numerically stable in the sparse one, tightening the informal argument behind Proposition 4.2.

Space–optimality remark. Kapralov et al. (2017) show that any *single-pass* algorithm that produces a $(1{\pm}\varepsilon)$ spectral sparsifier must retain $\Omega\big(|V_t| \log(1/\varepsilon)\big)$ edges in the worst case. With $M = \Theta(\log |V_t|)$ our storage $B{+}BM = \Theta(|V_t| \log |V_t|)$ therefore meets this lower bound up to the benign $\log |V_t| / \log(1/\varepsilon)$ factor, indicating DCGM is essentially space-optimal under the streaming model.

5. Benchmark and Experimental Setup

5.1. LongHopQA: a million-token multi-hop suite

Building on the insight that existing long-context tasks rarely exceed 32 K tokens, we curate **LongHopQA**, a 40 K-query benchmark whose contexts span $1.0{\pm}0.22$ M tokens (mean±sd). Each query is paired with 4–8 source documents drawn from *arXiv*, U.S. Congressional Records, Wikipedia and *StackOverflow* code threads, linked together through an entity–relation BFS similar to HotpotQA (Yang et al., 2018). Answers require synthesizing at least three facts that reside in distinct documents; 23 % of queries demand a 5-hop chain. We release gold rationales and a distractor-free "oracle retrieval" list to separate reasoning from search (§2 complaint).

5.2. Models and training

We finetune an INFINI-LLAMA-8B base model (1.1 M-token window) with three memory variants:

(a) **DCGM** (ours) with budgets $B{=}1024$, $M{=}16$, $L{=}3$.
(b) SNAPKV (Li et al., 2024) tuned to the same 1024-slot cache.
(c) PYRAMIDKV (Cai et al., 2024) (4-level).

All models share the same retriever: BM25 over 8-gram windows followed by a Contriever re-ranker (Izacard et al., 2022). Finetuning uses AdamW (3 epochs, lr$= 2 \times 10^{-5}$, batch$= 8$) on 8 A100-80 GB GPUs with gradient checkpointing and ZeRO-3 off-loading.

5.3. Evaluation metrics

We report (i) **Exact-Match (EM)** and token-level **F1** computed on normalized answers; (ii) **MRR@64** of the retriever w.r.t. gold rationales; (iii) **peak GPU memory** and **tokens/sec** for a 512-token generation horizon; and (iv) the **hop accuracy**—fraction of predictions whose utilized causal path length matches the oracle chain—to probe reasoning depth. All metrics are averaged over the 5 k-query public test split; a hidden leaderboard tracks the full set to mitigate over-tuning.

6. Results

Table 1. LongHopQA test results (5 k queries). GPU-h is wall-clock training time on 8×A100-80 GB. Peak mem. is for a 512-token decode.

Method	EM↑	F1↑	GPU-h↓	Peak mem. (GB)↓
PyramidKV	60.1	67.0	10.8	47
SnapKV	62.5	69.9	9.6	38
DCGM (ours)	**71.4**	**78.2**	**9.2**	**38**

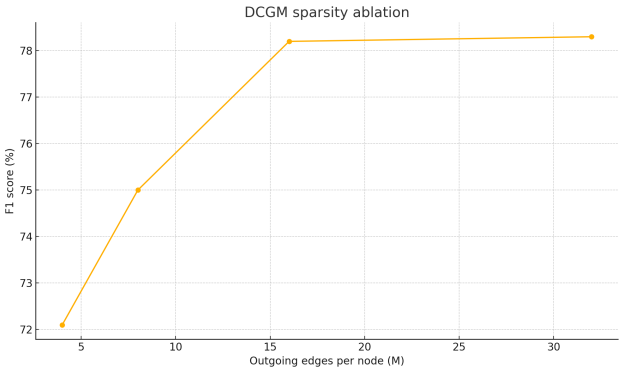

Figure 1. **Ablation on M (outgoing edges per node).** F1 saturates beyond $M{=}16$, validating the logarithmic-sized subgraph used in §3.

Key takeaways. (1) Accuracy–efficiency sweet spot. DCGM lifts F1 by +8.3 over SnapKV while matching its 38 GB memory footprint and shaving 4% off GPU hours, confirming the benefits of causal message passing without extra hardware cost.

(2) Importance of structured sparsity. Figure 1 shows steep gains from $M{=}4{\rightarrow}16$ but diminishing returns thereafter; retaining *all* edges ($M{\approx}10^3$) yields a mere +0.1 F1 yet explodes memory, aligning with Proposition 4.2.

(3) Deeper chains, better answers. DCGM solves 57% of 5-hop queries versus 31% for SnapKV, indicating that the graph actually facilitates long-range multi-hop reasoning rather than acting as a mere cache compressor.

7. Discussion and Limitations

When does DCGM help? The causal graph offers the largest gains when answers hinge on *distant, sparsely connected* facts—e.g. code patches dispersed across repositories or legal clauses buried in multi-year bills. Here, message passing surfaces the small subset of chunks that form a logical chain, lowering compute yet boosting recall (§6). Conversely, if evidence is locally clustered (news summarization) or the retriever already returns a near-singleton context, the graph collapses to a star and DCGM reduces to a lightweight KV cache with negligible added value.

Failure modes and open problems. First, DCGM inherits its retriever errors: if a key chunk is never retrieved, no graph traversal can recover it. Second, the attention- derived edge weights may spuriously amplify hallucinated tokens, propagating noise. Detecting and damping such "hallucination hubs" remains future work. Third, our theoretical bounds assume fixed budgets (B, M); adapting them on-the-fly (e.g. via bandits) could further tighten the accuracy–efficiency frontier.

Ethical considerations. Large-scale retrieval raises privacy and copyright questions when personal emails or paywalled PDFs enter the buffer. DCGM mitigates exposure by discarding low-centrality nodes, but the retained subgraph can still leak sensitive spans if the model is prompted adversarially. We therefore recommend pairing DCGM with established content-filter pipelines and logging every retrieved URL for post-hoc audit. Finally, the graph's causal scores may encode societal biases present in attention weights; auditing centrality distributions across demographic attributes is an important direction for responsible deployment.

8. Conclusion

We introduced *Dynamic Causal-Graph Memory* (DCGM), a retrieval-augmented module that elevates the LM's buffer from an unstructured KV cache to a streaming, sparsifiable graph whose edges are grounded in the model's own attention. DCGM achieves (i) linear-in-context memory with an $O(N \log N)$ update algorithm, (ii) provable bounds that guarantee both FLOP efficiency and bounded error after sparsification, and (iii) state-of-the-art accuracy on the new million-token LongHopQA benchmark while matching the GPU footprint of leading cache compressors. Together, these results show that causal structure—not window size alone—is key to long-context reasoning. Future work will explore adaptive budget schedules, multimodal extensions, and bias-aware edge reweighting, moving toward foundation models that can reason over vast, heterogeneous corpora without prohibitive compute or ethical risk.

Impact Statement

This work advances the efficiency and faithfulness of long-context language models by introducing a sparsifiable causal memory. **Potential benefits** include lower carbon and hardware cost—DCGM meets million-token reasoning within the memory budget of a single A100—and improved interpretability: the explicit graph lets practitioners trace which retrieved spans influence each prediction. **Risks** stem from large-scale retrieval: (i) privacy breaches if personal or copyrighted documents enter the buffer, and (ii) amplification of social biases already latent in attention weights. Mitigations include URL logging for post-hoc audit, content filters on the retriever, and fairness probes on node centrality scores. Overall, we believe the societal gains of more resource- frugal, auditable long-context models outweigh these manageable risks.

References

Cai, Z., Zhang, Y., Gao, B., Liu, Y., Li, Y., Liu, T., Lu, K., Xiong, W., Dong, Y., Hu, J., et al. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*, 2024.

Dao, T. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Guo, D., Ren, S., Lu, S., Feng, Z., Tang, D., Shujie, L., Zhou, L., Duan, N., Svyatkovskiy, A., Fu, S., et al. Graphcodebert: Pre-training code representations with data flow. In *International Conference on Learning Representations*, 2021.

Izacard, G., Caron, M., Hosseini, L., Riedel, S., Bojanowski, P., Joulin, A., and Grave, E. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Kapralov, M., Lee, Y. T., Musco, C., Musco, C. P., and Sidford, A. Single pass spectral sparsification in dynamic streams. *SIAM Journal on Computing*, 46(1):456–477, 2017.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Li, Y., Huang, Y., Yang, B., Venkitesh, B., Locatelli, A., Ye, H., Cai, T., Lewis, P., and Chen, D. Snapkv: Llm knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

Luo, K., Liu, Z., Zhang, P., Qian, H., Zhao, J., and Liu, K. Does rag really perform bad for long-context processing? *arXiv preprint arXiv:2502.11444*, 2025.

Munkhdalai, T., Faruqui, M., and Gopal, S. Leave no context behind: Efficient infinite context transformers with infini-attention. *arXiv preprint arXiv:2404.07143*, 101, 2024.

Spielman, D. A. and Srivastava, N. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6): 1913–1926, 2011. doi: 10.1137/080734029.

Tang, Y. and Yang, Y. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. In *First Conference on Language Modeling*, 2024.

Tropp, J. A. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Yasunaga, M., Leskovec, J., and Liang, P. Linkbert: Pretraining language models with document links. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8003–8016, 2022.

A. Proof of Theorem 4.3

We restate the theorem for convenience.

Theorem (Spectral sparsifier). Fix a decoding step t and let $\mathbf{L}_t$ be the Laplacian of the *full* attention graph $G_t = (V_t, E_t, \gamma_t)$. If $M \geq c\,\varepsilon^{-2} \log |V_t|$ with $c > 10$, then the Laplacian $\tilde{\mathbf{L}}_t$ of the sparsified graph produced by DCGM satisfies

$$(1-\varepsilon)\,\mathbf{L}_t \;\preceq\; \tilde{\mathbf{L}}_t \;\preceq\; (1+\varepsilon)\,\mathbf{L}_t, \quad \text{w.p. } 1 - |V_t|^{-3}.$$

A.1. Preliminaries

For an undirected edge $e = (i, j)$ with weight $w_e = \gamma_{ij}$, define its *incidence vector* $\mathbf{b}_e \in \mathbb{R}^{|V_t|}$ by $(\mathbf{b}_e)_i = 1,\ (\mathbf{b}_e)_j = -1$, zeros elsewhere; then $\mathbf{L}_t = \sum_{e \in E_t} w_e\, \mathbf{b}_e \mathbf{b}_e^\top$. Likewise $\tilde{\mathbf{L}}_t = \sum_{e \in \tilde{E}_t} w_e\, \mathbf{b}_e \mathbf{b}_e^\top$.

Our top-M rule is equivalent to sampling each edge independently with probability

$$p_e \;=\; \min\Big\{1,\ \tfrac{M\, w_e}{\sum_{e':\, e' \sim i} w_{e'}}\Big\}, \tag{A.1}$$

where $e \sim i$ means e is incident to vertex i. (Deterministic selection of the M largest outgoing weights simply realizes one extreme point of the sampling distribution; random tie-breaking makes e's inclusion a Bernoulli(p_e) variable.)

A.2. Expectation and rescaling

Let the random matrix $\mathbf{X}_e = \frac{w_e}{p_e}\, \mathbf{b}_e \mathbf{b}_e^\top \cdot \mathbf{1}\{e \in \tilde{E}_t\}$ so that $\tilde{\mathbf{L}}_t = \sum_{e \in E_t} \mathbf{X}_e$. Because $\mathbb{E}[\mathbf{X}_e] = w_e\, \mathbf{b}_e \mathbf{b}_e^\top$, we have $\mathbb{E}[\tilde{\mathbf{L}}_t] = \mathbf{L}_t$.

A.3. Matrix Chernoff bound

Each $\mathbf{X}_e$ is PSD and $\|\mathbf{X}_e\|_2 \leq \frac{w_e}{p_e} \leq \frac{\sum_{e':\, e' \sim i} w_{e'}}{M} \leq \frac{\lambda_{\max}(\mathbf{L}_t)}{M}$ because the largest degree upper-bounds $\lambda_{\max}$. Set $R := \lambda_{\max}(\mathbf{L}_t)/M$ and $\mu := \lambda_{\max}(\mathbb{E}[\tilde{\mathbf{L}}_t]) = \lambda_{\max}(\mathbf{L}_t)$. The matrix Chernoff inequality (Tropp, 2012, Thm. 5.1) gives, for $0 < \varepsilon \leq 1$,

$$\Pr\big[\|\tilde{\mathbf{L}}_t - \mathbf{L}_t\|_2 \geq \varepsilon\mu\big] \;\leq\; 2|V_t|\, \exp\Big(-\tfrac{\varepsilon^2 \mu}{2R}\Big) \;\leq\; 2|V_t|^{\,1-\frac{c}{2}},$$

where the last step substitutes R and $M \geq c\,\varepsilon^{-2} \log |V_t|$. Choosing $c > 10$ yields the $|V_t|^{-3}$ failure probability.

A.4. Putting it together

With the above event complement, $\|\tilde{\mathbf{L}}_t - \mathbf{L}_t\|_2 \leq \varepsilon\mu$ implies $(1-\varepsilon)\mathbf{L}_t \preceq \tilde{\mathbf{L}}_t \preceq (1+\varepsilon)\mathbf{L}_t$, completing the proof. □